

DCA0214.1 - LABORATÓRIO DE
ESTRUTURAS DE DADOS
Aula 9: Listas e suas generalizações; Árvores e
suas generalizações

Prof. Felipe Fernandes

03 Maio de 2019

- LEIA COM ATENÇÃO -

- I É proibido utilizar qualquer estrutura de dados ou algoritmos pré-fornecidos por alguma biblioteca C/C++.
- II Em cada aula desta unidade, as listas de exercícios possuirão questões marcadas com \star . Todas as questões (de todas as aulas) marcadas com \star , devem ser submetidas via SIGAA até às 23h59 do 16 Maio 2019. Peso: 20% na nota de segunda unidade.
- III O item II não lhe desobriga de resolver as questões sem \star . Todas as questões são importantes para a prova da segunda unidade.

-
- 1. Dada uma lista encadeada de caracteres formada por uma sequência alternada de letras e dígitos, construa um método que retorne uma lista na qual as letras são mantidas na sequência original e os dígitos são colocados na ordem inversa. Exemplos: $A1E5T7W8G$ deve retornar $AETWG8751$.
 - 2. Implemente uma função que recebe uma árvore binária T e retorna verdadeiro se T é binária de busca ou falso caso contrário. Complexidade requerida: $O(n)$
 - 3. Seja T uma árvore binária de busca qualquer, possuindo n nós. Implemente um algoritmo $O(n)$ que imprime as chaves de T em ordem crescente. Em seguida, implemente um algoritmo $O(n)$ que imprima as chaves de T em ordem decrescente.

4. Seja T uma árvore binária de busca qualquer, possuindo n nós. Seja $1 \leq k \leq n$ um inteiro positivo. Implemente uma função iterativa, com complexidade $O(n)$, que receba a raiz de T e o valor k e retorne o k -ésimo menor valor armazenado em T . (Dica: para obter o algoritmo iterativo, utilize uma pilha auxiliar).
5. Seja T uma árvore binária, possuindo n nós. Escreva um algoritmo iterativo, com complexidade $O(n)$, que retorne a quantidade folhas de T .
6. Seja T uma árvore binária qualquer. O percurso *in nível* visita os nós de T , nível a nível, começando no nível da raiz e terminando no nível das folhas. Em cada nível, o procedimento visita o nós da extrema esquerda até a extrema direita. Implemente um procedimento que percorre T *in nível*.
7. Seja T uma árvore binária qualquer. Dizemos que T' é espelho de T , quando T' é obtida trocando a subárvore direita pela subárvore esquerda de cada nó T . A Figura 1 ilustra uma árvore e seu espelho. Implemente uma função que receba T e retorne seu espelho.

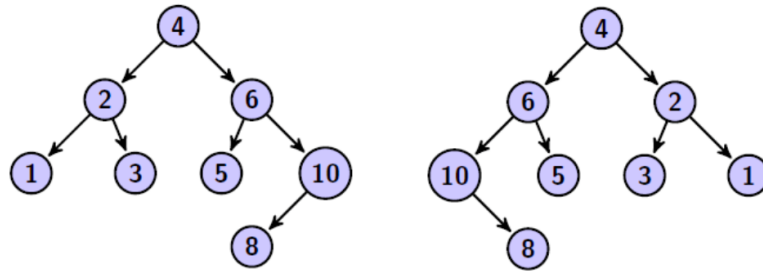


Figure 1: Espelho de uma árvore

8. Dada uma árvore binária de busca T qualquer, implemente duas funções eficientes: uma que retorne o comprimento de caminho interno e outra que retorne o comprimento de caminho externo de T .
9. ★ Seja $S = \{s_1, \dots, s_n\}$ um conjunto com n chaves numéricas. Seja $S' = \{s'_0, \dots, s'_n\}$ o conjunto de “chaves externas”, onde $s'_0 < s_1$, $s'_n > s_n$ e, $\forall i, 1 \leq i < n$, $s_i < s'_i < s_{i+1}$. Considere que lhe são dadas as probabilidades de acesso p_1, \dots, p_n de cada chave em S , bem como as probabilidades de acesso p'_0, \dots, p'_n de cada chave externa em S' . Implemente um algoritmo $O(n^3)$ que constrói a árvore de busca ótima e retorna seu custo final.